Vulnerability of Network Traffic under Node Capture Attacks using Circuit Theoretic Analysis

Patrick Tague*, David Slater*, Jason Rogers†, and Radha Poovendran*
*Network Security Lab (NSL), Electrical Engineering Department,
University of Washington, Seattle, Washington
†Center for High Assurance Computer Systems, Information Technology Division,
Naval Research Laboratory, Washington, DC
Email: {tague, dmslater, rp3}@u.washington.edu, rogers@itd.nrl.navy.mil

***Abstract*—We investigate the impact of node capture attacks on the confidentiality and integrity of network traffic. We map the compromise of network traffic to the flow of current through an electric circuit and propose a metric for quantifying the vulnerability of the traffic using the circuit mapping. We compute the vulnerability metric as a function of the routing and the cryptographic protocols used to secure the network traffic. We formulate the minimum cost node capture attack problem as a nonlinear integer programming problem. Due to the NP-hardness of the minimization problem, we provide a greedy heuristic that approximates the minimum cost attack. We provide examples of node capture attacks using our vulnerability metric and show that the adversary can expend significantly less resources to compromise target traffic by exploiting information leakage from the routing and cryptographic protocols.**

I. Introduction

The successful commercialization of many applications of wireless networks relies on the assurance of the *confidentiality* and *integrity* of the data communicated through the network. Confidentiality is defined as the ability to keep data secret from all but a set of authorized entities, and integrity is defined as the ability to verify that data has not been maliciously or accidentally altered while in transit [1]. Recent research has demonstrated that these properties can be efficiently compromised by physically capturing network nodes and extracting cryptographic keys from their memory [2]. Such *node capture attacks* are possible due to the potential unattended operation of wireless nodes and the prohibitive cost of tamper-proof hardware in portable devices [2]. Using the cryptographic keys recovered in a node capture attack, an adversary can compromise the confidentiality and integrity of any messages secured using the compromised keys.

Work of J. Rogers was performed while visiting the Network Security Lab at the University of Washington.

This work was supported in part by the following grants: ONR YIP, N00014-04-1-0479; ARO PECASE, W911NF-05-1-0491; ARL CTA; NSA/DoD IASP Fellowship; and ARO MURI, #W 911 NF 0710287.

This document was prepared through collaborative participation in the Communications and Networks Consortium sponsored by the US Army Research Laboratory under the Collaborative Technology Alliance Program, DAAD19-01-2-0011. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the US Government.

Recent literature [2]–[5] on symmetric key assignment [1] for resource-constrained devices has focused on node capture attacks in which an adversary chooses the captured nodes independently at random. In our previous work [6], [7], we showed that an intelligent adversary can reduce the resource expenditure required for the node capture attack using information leaked from the key assignment protocol. In particular, the adversary can learn which keys are assigned to individual nodes in the network by eavesdropping on or participating in the secure link establishment protocol [7].

For symmetric key assignment in wireless sensor and ad-hoc networks, node capture attacks aim at the compromise of individual node-to-node wireless links [2]–[7]. However, a message traversing multiple links between a source and destination node is compromised if any of the traversed links in the route becomes insecure. The overall security of a routed message is thus at best that of the least secure or most vulnerable link traversed by the message. Hence, the impact of the node capture attack is a function of both the cryptographic protocol which provides link security and the routing protocol which determines the links traversed by a given message.

In this paper, we use the vulnerability of network traffic as a measure of the adversary's ability to compromise a message traversing a particular route. By observing the network topology and inferring information from the routing and key assignment protocols, an intelligent adversary can analyze the vulnerability of traffic and capture the nodes which maximize the compromise of network traffic.

However, there is a resource expenditure associated with the capture of nodes and extraction of keys from their memory. Hence, the optimal attack strategy is that which captures a set of nodes with minimum total resource expenditure. This is in contrast to wiretapping attacks in routing or secure network coding [8], [9] which aim to tap a set of links with minimum total resource expenditure. An adversary with bounded resources will thus rely on an efficient node capture algorithm which minimizes the total resource expenditure. As we show in this paper, the joint consideration of information from the routing and key assignment protocols can lead to a significant reduction in resource expenditure compared to node capture attacks using routing or key assignment information separately.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
APR 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Vulnerability of Network Traffic under Node Capture Attacks using Circuit Theoretic Analysis	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Washington,Department of Electrical Engineering ,Network Security Lab (NSL),Seattle,WA,98195**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
27th IEEE Conference on Computer Communications (INFOCOM'08), 15-17 April 2008, Phoenix, AZ. U.S. Government or Federal Rights License

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**9**	

In this paper, we thus address the problem of *quantifying the minimum resources required in order to compromise the network traffic of a target set of source-destination pairs by jointly considering information from the routing and key assignment protocols*. Alternatively, we show the worst case impact of a node capture attack, given a specific amount of resources available to the adversary.

A. Our Contributions

The contributions of this work are summarized as follows.

- We map the compromise of network traffic to the flow of current through an electric circuit and derive a vulnerability metric using circuit analysis. Focusing on symmetric key distribution as in [2], [3], [6], we present a vulnerability metric for both single and multiple path routing topologies. Our proposed metric captures the gain achieved due to information leakage by the joint consideration of the routing and key assignment protocols.
- We formulate the minimum cost node capture attack problem as a nonlinear integer programming minimization problem using the proposed vulnerability metric. We present the GNAVE algorithm, a **G**reedy **N**ode capture **A**pproximation using **V**ulnerability **E**valuation, to approximate the minimum cost node capture attack.
- We demonstrate the impact of node capture attacks using the GNAVE algorithm in wireless networks with examples of both classical routing and network coding protocols. Furthermore, we compare the resource expenditure required for node capture attacks using the GNAVE algorithm to previously proposed strategies.

The remainder of this paper is organized is follows. In Section II, we present models and assumptions for the wireless network, key assignment, routing, and adversary. In Section III, we formulate the minimum cost node capture attack problem as a nonlinear integer programming minimization problem. We also derive a metric for the vulnerability of network traffic by mapping the compromise of messages to a circuit analysis problem. In Section IV, we analyze the node capture attack formulation with respect to the circuit analysis metric and present the GNAVE algorithm for node capture. In Section V, we present examples and simulation of node capture attacks for both classical routing and network coding protocols. In Section VI, we conclude and discuss future work.

II. Models and Notation

In this section, we introduce models for the wireless network, key assignment, routing, and adversary. We summarize the notation used in the paper in Table I.

A. Wireless Network Model

The network consists of a set $\mathcal{N}$ of wireless nodes. The network topology is represented as the directed *network graph* $G_n = (\mathcal{N}, L_n)$. The link set L_n represents the set of one-hop communicating neighbors and is equivalent to an asymmetric relation [10] such that each link (i, j), $i \neq j$, is in the link set L_n if and only if node i can reliably send a message to node j without intermediate relays.

TABLE I
A summary of notation is provided for reference.

Symbol	Definition
$\mathcal{N}$	Set of wireless nodes
L_n	Set of ordered pairs of one-hop neighbor nodes
G_n	Network graph $(\mathcal{N}, L_n)$
$\mathcal{K}_i, \mathcal{L}_i$	Sets of keys and labels assigned to node $i \in \mathcal{N}$
$\mathcal{K}_{ij}, \mathcal{L}_{ij}$	Sets of keys and labels shared by nodes i and j
$\mathcal{S}, \mathcal{D}$	Sets of source and destination nodes
$\mathcal{T}$	Subset of $\mathcal{S} \times \mathcal{D}$ of source-destination pairs
$\mathcal{T}_A$	Adversary's target set, subset of $\mathcal{T}$
$\mathcal{R}_{sd}$	Set of paths forming the route from s to d
f_π	Fraction of $\mathcal{R}_{sd}$ traffic traversing the path π
$G_n(s, d)$	Route subgraph of G_n corresponding to $\mathcal{R}_{sd}$
$\mathcal{C}$	Subset of $\mathcal{N}$ of captured nodes
$\mathcal{K}_\mathcal{C}, L_\mathcal{C}$	Sets of compromised keys and links when $\mathcal{C}$ captured
w_i	Weight or cost of capturing node $i \in \mathcal{N}$
$h_\mathcal{C}(s, d)$	Route vulnerability of $\mathcal{R}_{sd}$ when $\mathcal{C}$ captured
$\nu(i, \mathcal{C})$	Incremental value of node i when $\mathcal{C}$ captured
$R_\mathcal{C}(i, j)$	Link resistance of (i, j) when $\mathcal{C}$ captured
$R_\mathcal{C}(\mathcal{R}_{sd})$	Route resistance of $\mathcal{R}_{sd}$ when $\mathcal{C}$ captured

B. Key Assignment Model

We assume that there exist sets $\mathcal{K}$ of symmetric cryptographic keys and $\mathcal{L}$ of corresponding key labels. Each node $i \in \mathcal{N}$ is assigned a subset $\mathcal{K}_i$ of $\mathcal{K}$ and the corresponding subset $\mathcal{L}_i$ of $\mathcal{L}$. We denote the set of keys shared by nodes i and j as $\mathcal{K}_{ij} = \mathcal{K}_i \cap \mathcal{K}_j$ and allow communication between i and j if and only if $\mathcal{K}_{ij} \neq \varnothing$[1]. We assume that nodes i and j use the entire set $\mathcal{K}_{ij}$ of shared keys to secure the link (i, j), so the strength of the link security is directly related to the number of shared keys. We assume that each node i publicly broadcasts the label set $\mathcal{L}_i$, allowing each neighboring node j to determine the set $\mathcal{K}_{ij}$ of shared keys, as discussed in [2].

C. Routing Model

Let $\mathcal{S}$ and $\mathcal{D}$ respectively denote the subsets of $\mathcal{N}$ of source and destination nodes. The set of source-destination routing pairs is denoted as $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{D}$ and is constructed based on the decisions made by the routing protocol. A message from source $s \in \mathcal{S}$ to destination $d \in \mathcal{D}$ will traverse one or more directed *paths* determined by the routing protocol through the network graph G_n. Each routing path is defined as a set of sequential links (i, j) with $\mathcal{K}_{ij} \neq \varnothing$ connecting s and d in G_n. We define the *route* $\mathcal{R}_{sd}$ as the set of all paths traversed from s to d and the path weight f_π as the fraction of traffic in the route $\mathcal{R}_{sd}$ that traverses the path π.

The route $\mathcal{R}_{sd}$ can be represented graphically by the *route subgraph* $G_n(s, d)$ of G_n consisting only of nodes and directed links traversed by at least one path π in the route $\mathcal{R}_{sd}$.

We address three classes of routing protocols based on path multiplicity and dependence of messages being routed along different paths. The first class of protocols yield routes

[1]This requirement can be strengthened as in [3] to require $|\mathcal{K}_{ij}| \geq q$ for a fixed $q \geq 1$, though we do not explicitly address this requirement.

consisting of a single, fixed path, as in AODV or DSR [11] in a static network. The second class of protocols yield routes consisting of *multiple independent paths*, such that each message traverses a potentially different path, as in GBR or GEAR [11]. The third class of protocols yield routes consisting of *multiple dependent paths* used concurrently such that each message is coded or fragmented into multiple packets, each of which traverses a separate (not necessarily disjoint) path. This class contains, for example, protocols based on threshold secret sharing [12] and network coding [8], [9], [13] in which a set of coded packets must be decoded in order to recover the original message.

D. Adversarial Model

We assume that the adversary is bounded to polynomial-time computation and has sufficient but bounded resources to eavesdrop on and record messages throughout the network, capture nodes, and extract cryptographic keys from the memory of captured nodes. We assume the adversary has knowledge of the key assignment and routing protocols, including the route $\mathcal{R}_{sd}$ for each $(s,d) \in \mathcal{T}$ and the key label set $\mathcal{L}_i$ for each node i.

We assume that the primary goal of the adversary is to compromise all traffic of source-destination pairs in the *target set* $\mathcal{T}_A \subseteq \mathcal{T}$ by extracting cryptographic keys from the memory of captured nodes $\mathcal{C} \subseteq \mathcal{N}$ with minimum resource expenditure. The adversary thus captures nodes intelligently using the individual weight or *cost* w_i associated with the capture of and extraction of keys from the node i [7]. We do not address additional attacks following the node capture attack using the recovered keys.

III. Attack and Vulnerability Formulation

In this section, we formulate the minimum cost node capture attack problem as a nonlinear integer programming minimization problem. We map the compromise of network traffic to the flow of current through an electric circuit. Using the circuit mapping, we formally define the vulnerability of traffic traversing the route $\mathcal{R}_{sd}$.

A. Node Capture Attack Formulation

In order to evaluate the effect of capturing the nodes on the route $\mathcal{R}_{sd}$, we first provide definitions for the compromise of traffic due to the capture of nodes in $\mathcal{C}$. We denote the set of keys recovered by the adversary in capturing the subset $\mathcal{C}$ as $\mathcal{K}_\mathcal{C} = \bigcup_{i \in \mathcal{C}} \mathcal{K}_i$.

Definition 1: Any message which traverses the link $(i,j) \in L_n$ is *compromised* if $\mathcal{K}_{ij} \subseteq \mathcal{K}_\mathcal{C}$. We define the set $L_\mathcal{C} \subset L_n$ to be the set of such *compromised links*.

Using Definition 1, we further define the compromise of paths and message routes as follows.

Definition 2: The path π is compromised if there is at least one compromised link (i,j) in π.

Definition 3: The route $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$ is compromised if every path $\pi \in \mathcal{R}_{sd}$ is compromised.

According to Definition 3, any message sent from s to d is compromised if the route $\mathcal{R}_{sd}$ is compromised. Hence, to compromise all traffic routed between source-destination pairs in the target set $\mathcal{T}_A$, the adversary must choose a subset $\mathcal{C}$ that leads to the compromise of each route $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$. The choice of subset $\mathcal{C}$ requiring the minimum resource expenditure is thus given by the following minimum cost node capture problem.

Problem: Minimum Cost Node Capture Attack	
Given:	$\mathcal{L}_i, w_i$ for $i \in \mathcal{N}$ and $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$
Find:	$\mathcal{C} \subseteq \mathcal{N}$
such that	$\sum_{i \in \mathcal{C}} w_i$ is minimized
and	$\mathcal{R}_{sd}$ is compromised for all $(s,d) \in \mathcal{T}_A$.

B. Route Vulnerability Metric

Using Definition 3, an adversary can compute the fraction of target routes compromised due to the capture of a set of nodes $\mathcal{C}$. However, this fraction does not show how the set $\mathcal{C}$ should be selected. Furthermore, the fraction of compromised target routes does not capture the contribution of nodes in $\mathcal{C}$ toward the compromise of additional routes, as the compromise of a route is a binary event.

To adequately capture the progression toward the compromise of additional routes, we introduce the metric of route vulnerability $h_\mathcal{C}(s,d)$, defined as follows.

Definition 4: The *route vulnerability* $h_\mathcal{C}(s,d)$ of the route $\mathcal{R}_{sd}$ due to the capture of nodes in $\mathcal{C}$ is a quantity in the unit interval $[0,1]$ such that

1) $h_\varnothing(s,d) = 0$, where $\varnothing$ is the empty set,
2) $h_\mathcal{C}(s,d) = 1$ if and only if $\mathcal{R}_{sd}$ is compromised when $\mathcal{C}$ is captured, and
3) $h_{\mathcal{C}_1}(s,d) > h_{\mathcal{C}_2}(s,d)$ only if the capture of $\mathcal{C}_1$ is more beneficial to the adversary in compromising $\mathcal{R}_{sd}$ than the capture of $\mathcal{C}_2$.

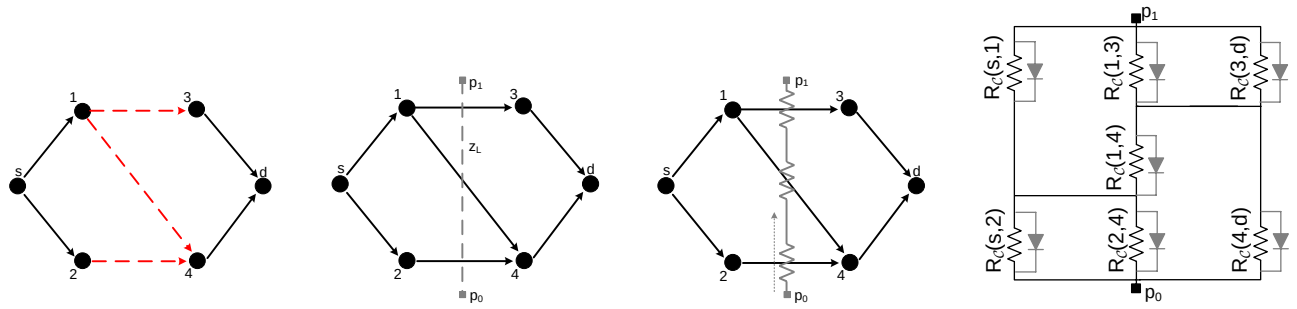
The metric of route vulnerability relaxes the binary notion of route compromise to a continuous measure of progress. Using the route vulnerability, we can devise a node capture strategy that maximizes the progression toward the goal of compromising all routes $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$. The final constraint in the minimum cost node capture attack problem can thus be replaced by the constraint $h_\mathcal{C}(s,d) = 1$ for all $(s,d) \in \mathcal{T}_A$.

While the boundary values of $h_\mathcal{C}(s,d)$ are well determined, condition 3) in Definition 4 does not quantify the intermediate values of $h_\mathcal{C}(s,d)$. In the next section, we define the intermediate values of $h_\mathcal{C}(s,d)$ using circuit theoretic analysis.

C. Mapping Route Compromise to Current Flow

In this section, we map the compromise of the route $\mathcal{R}_{sd}$ to the flow of current through an electric circuit and relate the route vulnerability $h_\mathcal{C}(s,d)$ to the resistance of the circuit. We first determine the compromise of a route $\mathcal{R}_{sd}$ according to the following Proposition.

Proposition 1: The route $\mathcal{R}_{sd}$ is compromised if and only if the set $L_\mathcal{C}$ of compromised links contains at least one (s,d) edge cut of the route subgraph $G_n(s,d)$ as a subset.



(a) (b) (c) (d)

Fig. 1. The figure illustrates the mapping from the compromise of the route $\mathcal{R}_{sd}$ to the flow of current through an electric circuit $\mathcal{E}_{sd}$. In (a), the route subgraph $G_n(s,d)$ is illustrated with the edge cut of compromised links indicated by dashed lines. In (b), the edge cut is replaced by the curve z_L directed from p_0 to p_1. In (c), the curve z_L is replaced by a wire, and a resistor is inserted in the wire at each point where the curve z_L crosses an edge (i,j) in $G_n(s,d)$. In (d), the circuit $\mathcal{E}_{sd}$ is illustrated by combining the wires and resistors for each possible edge cut L. The diode in parallel with each resistor maintains the orientation of edges in $G_n(s,d)$.

Proof: Suppose that $L_{\mathcal{C}}$ contains an edge cut of $G_n(s,d)$. By the definition of an edge cut, every path π from s to d in $G_n(s,d)$ necessarily passes through at least one link in the edge cut. By Definition 2, every path π in $\mathcal{R}_{sd}$ is compromised, implying by Definition 3 that the route itself is compromised.

Next, suppose that $\mathcal{R}_{sd}$ is compromised. Then, by Definition 2 and Definition 3, there is at least one compromised link (i_π, j_π) in each path $\pi \in \mathcal{R}_{sd}$, so let $L = \{(i_\pi, j_\pi) : \pi \in \mathcal{R}_{sd}\} \subseteq L_{\mathcal{C}}$. Since every path π traverses at least one edge in L, L is an edge cut of $G_n(s,d)$. ■

Proposition 1 thus implies that the task of compromising the route can be reduced to that of compromising an edge cut of $G_n(s,d)$ by capturing the set of nodes $\mathcal{C}$. We thus show that the compromise of the edge cut L of $G_n(s,d)$ is equivalent to the flow of current along a path through an electric circuit. The mapping is described by the following sequence of steps and illustrated in Fig. 1.

Step 1: The edge cut L is illustrated as a continuous, directed curve z_L which *crosses* $G_n(s,d)$ [14], crossing only the edges in L in a direction perpendicular to each edge. The set of compromised edges forming the edge cut L in Fig. 1(a) thus corresponds to the curve z_L in Fig. 1(b).

Step 2: The curve z_L crossing $G_n(s,d)$ is mapped to a wire carrying electric current from the starting point p_0 to the ending point p_1 of z_L. To represent the cost associated with the capture of nodes in $\mathcal{C}$ to compromise the edge cut L, a resistor of resistance $R_{\mathcal{C}}(i,j)$ is inserted at the point in the wire where the curve z_L crosses the edge $(i,j) \in L$. The curve z_L in Fig. 1(b) thus maps to the resistive current path from p_0 to p_1 in Fig. 1(c).

Step 3: The resistive current paths for each edge cut L of the graph $G_n(s,d)$ are then combined into an electric circuit $\mathcal{E}_{sd}$ with a single resistor of resistance $R_{\mathcal{C}}(i,j)$ corresponding to each edge (i,j) in $G_n(s,d)$. The circuit $\mathcal{E}_{sd}$ in Fig. 1(d) thus consists of all current paths from p_0 to p_1 such as that in Fig. 1(c). The cost of compromising the route $\mathcal{R}_{sd}$ is then proportional to the equivalent resistance of $\mathcal{E}_{sd}$. Since the resistors in $\mathcal{E}_{sd}$ are in one-to-one correspondence with the edges in $G_n(s,d)$, the circuit is related to the dual graph of $G_n(s,d)$.

In certain cases, the orientation of edges in $G_n(s,d)$ can lead to inconsistencies between edge cuts and current paths. For example, consider the edge cut $L = \{(s,1),(4,d)\}$ of $G_n(s,d)$ in Fig. 1(a). If the direction of the edge $(1,4)$ is ignored, L is no longer an edge cut, as the path $\{(s,2),(2,4),(4,1),(1,3),(3,d)\}$ is not compromised. Hence, the mapping must incorporate edge orientation. For example, in Fig. 1(a), the corresponding circuit must be modified such that the current flow incurs a cost $R_{\mathcal{C}}(1,4)$ in traversing the resistor toward p_1 but no cost in the other direction. This can be achieved by inserting an ideal diode in parallel with the resistor. Hence, the circuit mapping is completed by inserting an ideal diode in parallel with each resistor in the circuit according to the edge direction in $G_n(s,d)$, as in Fig. 1(d).

We note that the final step of combining resistive current paths into an electric circuit in Step 3 is well-defined only if the graph $G_n(s,d)$ with the additional edge (d,s) is a *planar graph*, i.e. only if $G_n(s,d)$ with (d,s) such that no edges intersect. This is due to the fact that the mapping above yields an electric circuit obtained as the *planar graph dual* [14] of $G_n(s,d)$ with (d,s). Hence, an alternate approach is required when $G_n(s,d)$ with (d,s) is not a planar graph. For example, if the edge $(2,3)$ is added to the graph $G_n(s,d)$ in Fig. 1(a), the resulting route cannot be analyzed using the planar graph dual.

To overcome the lack of a graph dual for non-planar graphs, we provide a mapping using the *circuit dual* [15] based on the duality of components and parameters in circuit analysis. In particular, the circuit $\mathcal{E}_{sd}$ can be constructed directly from $G_n(s,d)$ by replacing each directed edge (i,j) by a resistor of resistance $R_{\mathcal{C}}(i,j)^{-1}$ and a parallel diode allowing current to flow from j to i. The cost of compromising the route $\mathcal{R}_{sd}$ is then inversely proportional to the equivalent resistance of $\mathcal{E}_{sd}$. The circuit $\mathcal{E}_{sd}$ for the non-planar case resulting from the addition of the edge $(2,3)$ to $G_n(s,d)$ in Fig. 1(a) is illustrated

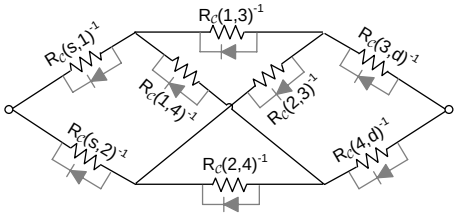


Fig. 2. The route subgraph $G_n(s,d)$ in Fig. 1(a) is made non-planar by adding the edge $(2,3)$. The resulting electric circuit $\mathcal{E}_{sd}$ is illustrated.

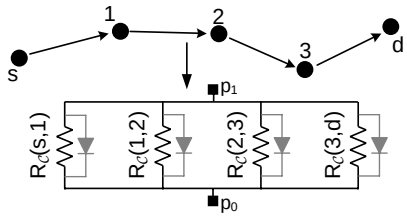


Fig. 3. The route resistance $R_{\mathcal{C}}(\{\pi\})$ of the single path route $\mathcal{R}_{sd} = \{\pi\}$ is given by (1) as the equivalent resistance of the parallel resistors $R_{\mathcal{C}}(i,j)$.

in Fig. 2. We next show how circuit analysis techniques can be used in computing the route vulnerability $h_{\mathcal{C}}(s,d)$ from the electric circuit $\mathcal{E}_{sd}$.

D. A Measure of Vulnerability of Network Traffic

In this section, we define the route vulnerability $h_{\mathcal{C}}(s,d)$ of the route $\mathcal{R}_{sd}$ as a function of the equivalent resistance of the electric circuit $\mathcal{E}_{sd}$. We first provide a definition of the resistance values $R_{\mathcal{C}}(i,j)$ for the resistors in the circuit $\mathcal{E}_{sd}$.

Definition 5: The *link resistance* $R_{\mathcal{C}}(i,j)$ of the resistor in $\mathcal{E}_{sd}$, for both the planar and non-planar circuit mappings, is equal to the number of keys securing the link (i,j) that are not compromised and is given by $R_{\mathcal{C}}(i,j) = |\mathcal{K}_{ij} \setminus \mathcal{K}_{\mathcal{C}}|$.

We note that the link resistance values are a measure of the resilience of individual links to the capture of nodes in $\mathcal{C}$. In the planar circuit mapping, the overall resistance of the circuit $\mathcal{E}_{sd}$ is thus *inversely proportional* to the ease with which the adversary compromises the route $\mathcal{R}_{sd}$. In contrast, the overall resistance of the circuit $\mathcal{E}_{sd}$ in the non-planar circuit mapping is *directly proportional* to the ease with which the adversary compromises $\mathcal{R}_{sd}$. We thus provide the following definition for the resistance of the circuit $\mathcal{E}_{sd}$ in each case.

Definition 6: The *route resistance* $R_{\mathcal{C}}(\mathcal{R}_{sd})$ is defined as the resilience of the route $\mathcal{R}_{sd}$ to the capture of nodes in $\mathcal{C}$. In the planar circuit mapping, $R_{\mathcal{C}}(\mathcal{R}_{sd})$ is the equivalent resistance of the circuit $\mathcal{E}_{sd}$. In the non-planar circuit mapping, $R_{\mathcal{C}}(\mathcal{R}_{sd})^{-1}$ is the equivalent resistance of the circuit $\mathcal{E}_{sd}$.

We note that the link resistance is a function only of the key assignment protocol, while the route resistance is a function of both the key assignment and routing protocols.

We next define the route vulnerability $h_{\mathcal{C}}(s,d)$ as a function of the route resistance $R_{\mathcal{C}}(\mathcal{R}_{sd})$. Definitions 5 and 6 imply that a link or route is more vulnerable to attack when the link or route resistance is smaller, in the same manner that current flows easier through a path of smaller resistance. Hence, we define the route vulnerability to be inversely proportional to the route resistance. To satisfy the conditions of Definition 4, the route vulnerability $h_{\mathcal{C}}(s,d)$ is defined as follows.

Definition 7: The route vulnerability is defined as

$$h_{\mathcal{C}}(s,d) = \frac{1}{R_{\varnothing}\left(\mathcal{R}_{sd}\right)}\left(\frac{1+R_{\varnothing}\left(\mathcal{R}_{sd}\right)}{1+R_{\mathcal{C}}\left(\mathcal{R}_{sd}\right)}-1\right),$$

where $\varnothing$ denotes the empty set. Conditions 1) and 2) in Definition 4 are trivially satisfied by this definition, and condition 3) is satisfied by noting that $h_{\mathcal{C}}(s,d)$ is inversely proportional to the route resistance $R_{\mathcal{C}}(s,d)$ which measures the resilience of the route to the capture of nodes in $\mathcal{C}$.

The evaluation of the route vulnerability $h_{\mathcal{C}}(s,d)$ varies between single, multiple independent, and multiple dependent path routing protocols as a function of the equivalent resistance of the circuit $\mathcal{E}_{sd}$. Hence, we address the three cases separately by denoting the route vulnerability as $h^{\pi}_{\mathcal{C}}(s,d)$, $h^{I}_{\mathcal{C}}(s,d)$, and $h^{D}_{\mathcal{C}}(s,d)$ for single, multiple independent, and multiple dependent path protocols, respectively.

For single path routing protocols, the route $\mathcal{R}_{sd}$ is given by a single directed path π from the source s to the destination d. The circuit mapping using the planar graph dual can thus be applied. As illustrated by Fig. 3, the equivalent circuit $\mathcal{E}_{sd}$ for $\mathcal{R}_{sd} = \{\pi\}$ is a parallel combination of link resistors $R_{\mathcal{C}}(i,j)$ for $(i,j) \in \pi$. Hence, the route resistance $R_{\mathcal{C}}(\{\pi\})$ is given by

$$R_{\mathcal{C}}(\{\pi\}) = \left(\sum_{(i,j)\in\pi} R_{\mathcal{C}}(i,j)^{-1}\right)^{-1}. \tag{1}$$

The route vulnerability $h^{\pi}_{\mathcal{C}}(s,d)$ is thus given by Definition 7 and (1) as

$$h^{\pi}_{\mathcal{C}}(s,d) = \frac{1}{R_{\varnothing}\left(\{\pi\}\right)}\left(\frac{1+R_{\varnothing}\left(\{\pi\}\right)}{1+R_{\mathcal{C}}\left(\{\pi\}\right)}-1\right). \tag{2}$$

In the case of multiple independent path routing protocols, the adversary can compromise messages traversing individual paths without compromising the route. In particular, the compromise of each path $\pi \in \mathcal{R}_{sd}$ yields the compromise of a fraction f_{π} of the traffic from s to d. Hence, the route vulnerability $h^{I}_{\mathcal{C}}(s,d)$ can be computed using (1) and (2) as

$$h^{I}_{\mathcal{C}}(s,d) = \sum_{\pi\in\mathcal{R}_{sd}} f_{\pi} h^{\pi}_{\mathcal{C}}(s,d). \tag{3}$$

For multiple dependent path routing protocols, messages from s to d are compromised only when the entire route $\mathcal{R}_{sd}$ is compromised. Hence, the route vulnerability $h^{D}_{\mathcal{C}}(s,d)$ is given directly by Definition 7 using the equivalent resistance $R_{\mathcal{C}}(\mathcal{R}_{sd})$.

IV. A Heuristic Node Capture Algorithm using Route Vulnerability

Given the definition of route vulnerability metric $h_{\mathcal{C}}(s,d)$ derived in Section III-D, we now propose a heuristic algorithm which iteratively captures those nodes which maximize the increase in route vulnerability.

Based on the definition of path compromise in Definition 2 and the circuit analysis techniques used to define the route vulnerability, the metric $h_{\mathcal{C}}(s,d)$ is nonlinear in the entries of

$\mathcal{C}$. Hence, the minimum cost node capture attack formulated in Section III-A is a nonlinear integer programming minimization problem.

Due to the fact that integer programming minimization is an NP-hard problem [10], [16] and because of the nonlinearity of $h_{\mathcal{C}}(s,d)$, we propose the use of a greedy heuristic that iteratively adds nodes to $\mathcal{C}$ based on the increase in route vulnerability $h_{\mathcal{C}}(s,d)$. The heuristic is thus similar to a known greedy heuristic for set covering [17] and linear integer programming [16]. However, due to the nonlinearity in $h_{\mathcal{C}}(s,d)$, the worst-case performance of the greedy heuristic cannot be analyzed using the ratio bound analysis in [10], [16], [17] and is left as an open problem for our future research.

Though any appropriate heuristic will eventually lead to the compromise of all routes $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$, it may be beneficial to the adversary to attempt to maximize the vulnerability resulting from the capture of each individual node using the information recovered from previously captured nodes. The contribution of a node i is given by the increase in route vulnerability $h_{\mathcal{C}\cup\{i\}}(s,d) - h_{\mathcal{C}}(s,d)$ due to the addition of i to $\mathcal{C}$, weighted by the adversary's preference for $\mathcal{R}_{sd}$ over other routes, indicated by a non-negative weight v_{sd}. The value of each node i is thus defined as follows.

Definition 8: The individual *incremental node value* of adding node $i \in \mathcal{N}$ to $\mathcal{C}$ is defined as

$$\nu(i,\mathcal{C}) = \sum_{(s,d)\in\mathcal{T}_A} v_{sd}\left(h_{\mathcal{C}\cup\{i\}}(s,d) - h_{\mathcal{C}}(s,d)\right).$$

To maximize the cost-effectiveness of the node capture attack at each iteration, the adversary chooses to capture the node with maximum value per unit cost $\nu(i,\mathcal{C})/w_i$. Based on this greedy approach, we propose the GNAVE algorithm, where GNAVE stands for **G**reedy **N**ode capture **A**pproximation using **V**ulnerability **E**valuation.

GNAVE Algorithm

Given: $\mathcal{L}_i, w_i$ for $i \in \mathcal{N}$, $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$
$\mathcal{C} \leftarrow \varnothing$
while there exists $(s,d) \in \mathcal{T}_A$ with $h_{\mathcal{C}}(s,d) < 1$ **do**
  $i^* \leftarrow \arg\max_{i\in\mathcal{N}} \nu(i,\mathcal{C})/w_i$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{i^*\}$
end while

Each iteration of the GNAVE algorithm is executed as follows. The adversary constructs the electric circuit $\mathcal{E}_{sd}$ for a given route $\mathcal{R}_{sd}$ and set $\mathcal{C}$ of captured nodes in order to compute the route vulnerability $h_{\mathcal{C}}(s,d)$. For each node $i \in \mathcal{N} \setminus \mathcal{C}$, the circuit $\mathcal{E}_{sd}$ is then modified by updating the link resistance values with respect to the keys that would be compromised if node i was captured. The potential route vulnerability $h_{\mathcal{C}\cup\{i\}}(s,d)$ is then computed using the equivalent resistance of the modified circuit. The increase in route vulnerability is aggregated over all routes $\mathcal{R}_{sd}$ for $(s,d) \in \mathcal{T}_A$, yielding the node value $\nu(i,\mathcal{C})$ as in Definition 8. The impact of the GNAVE algorithm is demonstrated in Section V through examples.

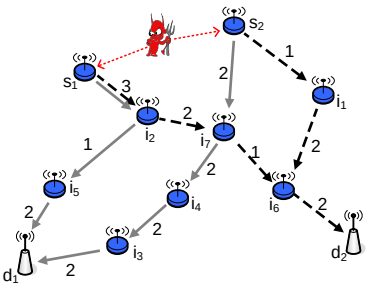


Fig. 4. Sources s_j sense an event and send reports to the destinations d_j via single path routes using relay nodes i_j. Each link (i,j) is labeled with the number of shared keys $|\mathcal{K}_{ij}|$.

V. Case Study

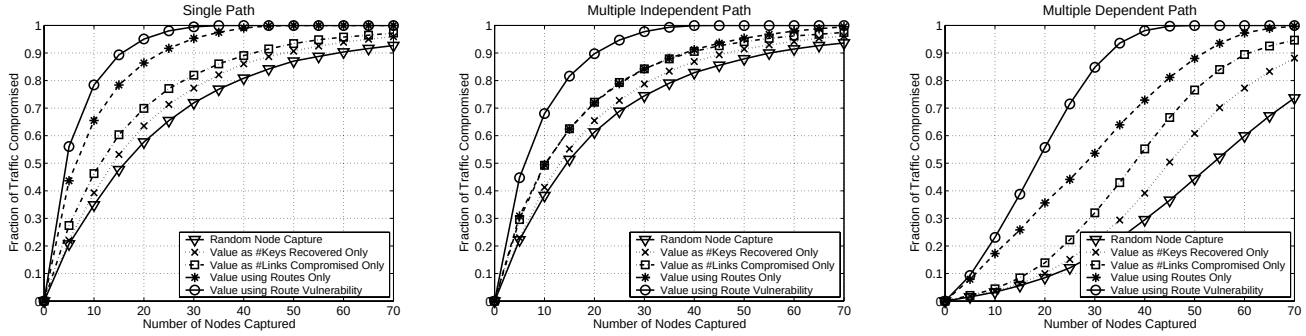
In this section, we illustrate the application of the route vulnerability metric $h_{\mathcal{C}}(s,d)$ and the GNAVE algorithm via two examples. We first analyze attacks on a single path routing protocol in a wireless sensor network using key predistribution [2], [3], [6]. We then analyze attacks on a randomized network coding protocol [13] in a static wireless network using symmetric broadcast keys. For each case, we illustrate a node capture attack in detail and provide simulation results for a large-scale network. In both cases, we assume that the source and relay nodes are low-level sensor nodes and, hence, set the node capture cost w_i for each source s_j and relay i_j to unity. We similarly assume the sink nodes are higher-level devices equipped with tamper-resistant hardware and, hence, set the node capture cost w_i for each destination d_j to infinity. Under these assumptions, the metric of minimum cost reduces to the minimum number of captured nodes.

For the large-scale network simulations, we compare the results of the following five node capture strategies.

1) Nodes are captured independently at random, serving as the baseline performance for the adversary.
2) Nodes are captured iteratively to maximize the number of compromised keys $|\mathcal{K}_{\mathcal{C}}|$. The node i with maximum $|\mathcal{K}_i \setminus \mathcal{K}_{\mathcal{C}}|$ is captured using information leaked from the key assignment protocol [7].
3) Nodes are captured iteratively to maximize the number of compromised links $|L_{\mathcal{C}}|$. The node i which compromises the maximum number of additional links is captured using information leaked from the key assignment protocol [7].
4) Nodes are captured iteratively to maximize the amount of network traffic routed through captured nodes.
5) Nodes are captured using the GNAVE algorithm and the route vulnerability metric $h_{\mathcal{C}}(s,d)$, using information from both the routing and key assignment protocols.

A. Key Predistribution in Wireless Sensor Networks

We first evaluate the route vulnerability for routes in a wireless sensor network using a single path routing protocol with keys assigned to sensor nodes prior to deployment using key predistribution [2], [3], [6]. In this example, we address the sensor network topology given in Fig. 4 with keys distributed to nodes as follows.



(a) (b) (c)

Fig. 5. The five node capture strategies are illustrated for a wireless sensor network of $|\mathcal{N}| = 500$ nodes for (a) single path, (b) multiple independent path, and (c) multiple dependent path routing.

$\mathcal{K}_{s_1} = \{k_2, k_3, k_5, k_7\}$	$\mathcal{K}_{s_2} = \{k_1, k_6, k_8, k_9\}$
$\mathcal{K}_{d_1} = \{k_1, k_2, k_3, k_4\}$	$\mathcal{K}_{d_2} = \{k_3, k_6, k_8, k_{10}\}$
$\mathcal{K}_{i_1} = \{k_2, k_4, k_8, k_{10}\}$	$\mathcal{K}_{i_2} = \{k_2, k_4, k_5, k_7\}$
$\mathcal{K}_{i_3} = \{k_1, k_3, k_7, k_{11}\}$	$\mathcal{K}_{i_4} = \{k_1, k_6, k_9, k_{11}\}$
$\mathcal{K}_{i_5} = \{k_1, k_4, k_8, k_{10}\}$	$\mathcal{K}_{i_6} = \{k_3, k_4, k_7, k_8\}$
$\mathcal{K}_{i_7} = \{k_4, k_5, k_6, k_9\}$	

To demonstrate how each link is secured using the assigned keys $\mathcal{K}_i$, we note that nodes i_2 and i_7 share keys $\mathcal{K}_{i_2 i_7} = \{k_4, k_5\}$. The link (i_2, i_7) is thus secure as long as $\{k_4, k_5\} \nsubseteq \mathcal{K}_{\mathcal{C}}$.

The route resistance of each of the four source-destination routes illustrated in Fig. 4 can be computed using (1) with the link resistance of each link (i, j) given by Definition 5. Using the route vulnerability metric $h^{\pi}_{\mathcal{C}}(s, d)$ in (2) and the GNAVE algorithm, the first captured node is chosen by evaluating the incremental value $\nu(i, \varnothing)$ for each node i, provided in Table II.

To demonstrate the computation of the quantities in Table II, we consider the source-destination pair (s_2, d_1) in the fourth column of Table II. The route $\mathcal{R}_{s_2 d_1} = \{\pi\}$ consists of a single path $\pi = \{(s_2, i_7), (i_7, i_4), (i_4, i_3), (i_3, d_1)\}$, so the route resistance $R_{\varnothing}(\{\pi\})$ prior to the attack is equal to the parallel resistance of the four link resistors. As indicated in Fig. 4, each link resistor in π has a resistance of 2 when $\mathcal{C} = \varnothing$, so $R_{\varnothing}(\mathcal{R}_{s_2 d_1}) = 1/2$ by (1). If node i_5 is added to $\mathcal{C}$, the links in π will have corresponding link resistances $R_{\{i_5\}}(s_2, i_7) = 2$, $R_{\{i_5\}}(i_7, i_4) = 2$, $R_{\{i_5\}}(i_4, i_3) = 1$, and $R_{\{i_5\}}(i_3, d_1) = 1$ by Definition 5. Hence, by (1), $R_{\{i_5\}}(\{\pi\}) = 1/3$. The route vulnerability $h^{\pi}_{\{i_5\}}(s_2, d_1)$ is thus given by (2) as

$$h^{\pi}_{\{i_5\}}(s_2, d_1) = \frac{1}{1/2}\left(\frac{1 + 1/2}{1 + 1/3} - 1\right) = 1/4.$$

As indicated in Table II, the first node added to $\mathcal{C}$ using GNAVE is node i_5 with the value $\nu(i_5, \varnothing) = 3.25$. We note that the choice of node i_5 is not obvious given the routing topology in Fig. 4. In fact, based on the topology alone, it appears as though nodes i_2, i_6, and i_7 would all be better choices as two of the four routing paths traverse these nodes. However, when considering the compromise of non-incident links throughout the network due to the recovery of keys, the capture of i_5 is more beneficial to the adversary.

TABLE II
THE ROUTE RESISTANCE, ROUTE VULNERABILITY, AND NODE VALUE ARE COMPUTED FOR EACH NODE i IN THE NETWORK IN FIG. 4.

i	$R_{\{i\}}(\mathcal{R}_{sd})$, $h_{\{i\}}(s, d)$				$\nu(i, \varnothing)$
	(s_1, d_1)	(s_1, d_2)	(s_2, d_1)	(s_2, d_2)	
s_1	$0, 1$	$0, 1$	$2/5, 1/7$	$2/5, 1/7$	2.29
s_2	$3/7, 3/20$	$6/17, 3/23$	$0, 1$	$0, 1$	2.28
i_1	$0, 1$	$0, 1$	$1/2, 0$	$0, 1$	3
i_2	$0, 1$	$0, 1$	$1/2, 0$	$2/5, 1/7$	2.29
i_3	$1/3, 7/24$	$2/7, 7/27$	$0, 1$	$1/2, 0$	1.55
i_4	$3/7, 3/20$	$3/7, 0$	$0, 1$	$1/2, 0$	1.15
i_5	$0, 1$	$0, 1$	$\mathbf{1/3, 1/4}$	$0, 1$	$\mathbf{3.25}$
i_6	$0, 1$	$0, 1$	$2/5, 1/7$	$0, 1$	3.14
i_7	$0, 1$	$0, 1$	$0, 1$	$2/5, 1/7$	3.14

In order to observe the performance of node capture attacks in a large-scale wireless sensor network, we also simulated the five node capture attacks above for each of the three routing protocol classes: single path, multiple independent path, and multiple dependent path routing.

Each simulation was performed for a network of $|\mathcal{N}| = 500$ nodes with $|\mathcal{K}_i| = 50$ randomly selected keys for each node $i \in \mathcal{N}$ and deployed with an average of 25 neighbors. The subsets $\mathcal{S}, \mathcal{D} \subseteq \mathcal{N}$ were randomly selected such that $|\mathcal{S}| = 100$ and $|\mathcal{D}| = 10$. Each source node chose to route messages to the nearest three of the 10 destination nodes. In our simulation, we implemented geographic forwarding with a hop-count mechanism to avoid routing loops and geographic dead-ends [11]. For single path routing, the next hop neighbor was chosen as the neighbor closest to the destination with a lower or equal hop count, while for multiple (independent and dependent) path routing, three such neighbors were chosen. For multiple dependent path routing, we assume that any minimum edge cut is sufficient to reconstruct the original message.

Fig. 5 illustrates the node capture attacks on each of the three cases of single path, multiple independent path, and multiple dependent path routing. We note that the node capture attack using the GNAVE algorithm requires the capture of significantly fewer nodes for all three routing protocol classes

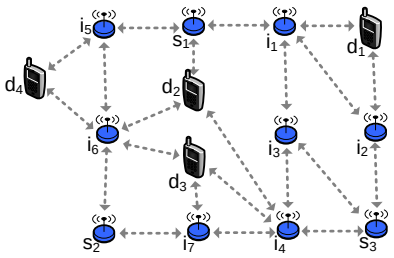


Fig. 6. The network and shared broadcast keys are illustrated for three sources s_1, s_2, and s_3 multicasting messages to groups $\{d_1, d_2, d_3, d_4\}$, $\{d_2, d_3, d_4\}$, and $\{d_1, d_2, d_3\}$, respectively. Each node is joined by edges to the set of neighbors which can receive secure broadcast transmissions.

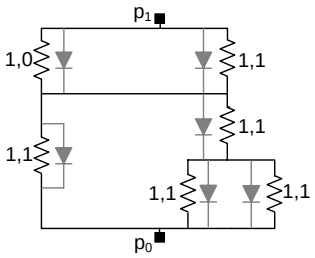


Fig. 7. The electric circuit $\mathcal{E}_{s_2 d_3}$ is illustrated for the route $\mathcal{R}_{s_2 d_3}$ in the network of Fig. 6. The label on each resistor provides the link resistances $R_{\mathcal{C}}(i,j)$ for both $\mathcal{C} = \varnothing$ and $\mathcal{C} = \{i_5\}$ for each link (i,j).

compared to the first four attacks. In comparing Fig. 5(b) and Fig. 5(c), we note that the dependence of messages traversing different paths prevents the quick increase in the compromise of traffic for a small number of captured nodes. However, the number of captured nodes $|\mathcal{C}|$ required to compromise all target traffic is only slightly increased. Hence, although multiple path routing is more resilient to the capture of a small number of nodes compared to single path routing, the same resource expenditure is required to compromise all traffic in all three cases.

To compare the five different node capture strategies, we compare the number of nodes required to compromise 80% of network traffic, approximately 40, 32, 27, 16, and 10 for the five attacks on single path routing. Hence, the total resource expenditure due to the capture of nodes in $\mathcal{C}$ using the route vulnerability metric $h_{\mathcal{C}}(s,d)$ is $25-65\%$ of that required by the first four simulated strategies.

B. Network Coding with Symmetric Broadcast Keys

We next evaluate the route vulnerability using randomized network coding combined with symmetric key encrypted broadcasts. A unique broadcast key is assigned to each node and a random subset of its neighbors. In this example, we address the network topology given in Fig. 6. The following periodic broadcast schedule demonstrates how broadcast messages propagate through the network using randomized network coding at each time t.

t	Sender	Message
0	s_1	x_1
	s_2	x_2
	s_3	x_3
1	i_5	$\alpha_5 x_1$
	i_1	$\alpha_1 x_1$
	i_7	$\alpha_7 x_2$
2	i_6	$\alpha_6 x_2 + \beta_6(\alpha_5 x_1)$
	i_3	$\alpha_3 x_3 + \beta_3(\alpha_1 x_1)$
3	i_2	$\alpha_2 x_3 + \beta_2(\alpha_1 x_1)$
	i_4	$\alpha_4 x_3 + \beta_4(\alpha_7 x_2) + \gamma_4(\alpha_3 x_3 + \beta_3(\alpha_1 x_1))$

Each message is broadcast, encrypted and authenticated with the corresponding keys, to each key sharing neighbor, as indicated in Fig. 6. The parameters α_i, β_i, and γ_i are randomly selected network coding coefficients chosen from a given finite field [13].

Since the example network in Fig. 6 is planar, the route resistance of each of the ten source to destination routes can be computed by constructing the corresponding circuit using the planar graph dual as illustrated in Section III-C. The use of distinct broadcast keys suggests that a single key is used to secure each directed link (i,j). Using the route vulnerability metric in Section III for dependent path routing protocols and the GNAVE algorithm, the adversary's choice for the first captured node is given by evaluating the incremental value $\nu(i,\varnothing)$ for each node i. The evaluation of node value $\nu(i,\varnothing)$ is given in Table III.

To demonstrate the computation of the quantities in Table III, we consider the source-destination pair (s_2, d_3) in the seventh column of the table and illustrate the computation of the route resistance $R_{\{i_5\}}(\mathcal{R}_{s_2 d_3})$ and the route vulnerability $h^{\pi}_{\{i_5\}}(s_2, d_3)$ due to the capture of node i_5. Based on the randomized network coding protocol, the route consists of the three paths $\pi_1 = \{(s_2, i_6), (i_6, d_3)\}$, $\pi_2 = \{(s_2, i_7), (i_7, d_3)\}$, and $\pi_3 = \{(s_2, i_7), (i_7, i_4), (i_4, d_3)\}$. The equivalent electric circuit for the route $\mathcal{R}_{s_2 d_3}$ is thus given in Fig. 7 with resistor values given for both $\mathcal{C} = \varnothing$ and $\mathcal{C} = \{i_5\}$.

Using resistive circuit evaluation, the equivalent resistance of the circuit in Fig. 7 is $R_{\varnothing}(\mathcal{R}_{s_2 d_3}) = 11/10$ for $\mathcal{C} = \varnothing$ and $R_{\{i_5\}}(\mathcal{R}_{s_2 d_3}) = 3/5$ for $\mathcal{C} = \{i_5\}$. The route vulnerability $h^{D}_{\{i_5\}}(s_2, d_3)$ is thus given by Definition 7 as

$$h^{D}_{\{i_5\}}(s_2, d_3) = \frac{1}{11/10}\left(\frac{1+11/10}{1+3/5} - 1\right) = 25/88.$$

As indicated in Table III, the first node added to $\mathcal{C}$ using the GNAVE algorithm is node i_5 with the value $\nu(i_5, \varnothing) = 5.73$.

We next simulated the performance of node capture attacks in a large-scale network using randomized network coding [13]. The implementation of randomized network coding in this example combines network coding with geographic flooding, in that coded packets are only propagated in the direction of the destination nodes using a hop count mechanism to avoid geographic dead-ends [11]. Broadcast keys were assigned randomly within each neighborhood, in that each neighbor of a node is in the key-sharing subset with probability p computed to guarantee a connected network with high probability [2], [6]. Each node in the network thus receives encrypted packets from upstream neighbors, decrypts each packet, computes a linear combination of the coded packets using random coefficients, and encrypts and forwards the resulting packet to downstream neighbors. Similar to the previous example,

TABLE III
THE ROUTE RESISTANCE, ROUTE VULNERABILITY, AND NODE VALUE ARE COMPUTED FOR EACH NODE i IN THE NETWORK IN FIG. 6.

i	$R_{\{i\}}(\mathcal{R}_{sd})$, $h_{\{i\}}(s,d)$										$\nu(i,\varnothing)$
	(s_1,d_1)	(s_1,d_2)	(s_1,d_3)	(s_1,d_4)	(s_2,d_2)	(s_2,d_3)	(s_2,d_4)	(s_3,d_1)	(s_3,d_2)	(s_3,d_3)	
s_1	$0,1$	$0,1$	$0,1$	$0,1$	$5/6,0$	$11/10,0$	$1/2,0$	$1/2,0$	$3/5,0$	$3/5,0$	4
s_2	$3/5,0$	$5/4,16/171$	$1/4,16/35$	$1/2,1/9$	$0,1$	$0,1$	$0,1$	$1/2,0$	$3/5,0$	$3/5,0$	3.66
s_3	$1/2,1/9$	$4/3,9/133$	$1/3,9/28$	$3/5,0$	$1/2,4/15$	$1,1/22$	$1/2,0$	$0,1$	$0,1$	$0,1$	3.81
i_1	$0,1$	$0,1$	$0,1$	$0,1$	$5/6,0$	$11/10,0$	$1/2,0$	$0,1$	$1/2,1/9$	$1/2,1/9$	5.22
i_2	$0,1$	$4/3,9/133$	$1/3,9/28$	$3/5,0$	$5/6,0$	$11/10,0$	$1/2,0$	$0,1$	$0,1$	$0,1$	4.39
i_3	$0,1$	$4/3,9/133$	$1/3,9/28$	$3/5,0$	$1/2,4/15$	$1,1/22$	$1/2,0$	$0,1$	$0,1$	$0,1$	4.70
i_4	$3/5,0$	$4/3,9/133$	$1/3,9/28$	$3/5,0$	$1/2,4/15$	$1/2,4/11$	$1/2,0$	$0,1$	$0,1$	$0,1$	4.02
i_5	$0,1$	$0,1$	$0,1$	$0,1$	$1/3,9/20$	$\mathbf{3/5,25/88}$	$0,1$	$1/2,0$	$3/5,0$	$3/5,0$	$\mathbf{5.73}$
i_6	$3/5,0$	$5/4,16/171$	$1/4,16/35$	$0,1$	$0,1$	$0,1$	$0,1$	$1/2,0$	$3/5,0$	$3/5,0$	4.55
i_7	$3/5,0$	$4/3,9/133$	$1/3,9/28$	$3/5,0$	$0,1$	$0,1$	$0,1$	$1/2,0$	$0,1$	$0,1$	5.39

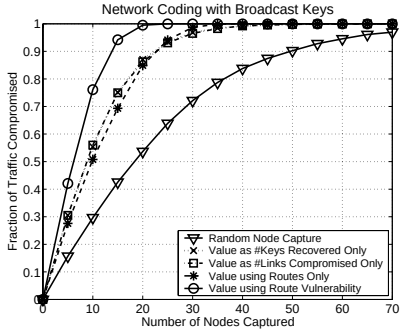


Fig. 8. Node capture attacks are performed using five node capture strategies for a randomized network coding protocol using broadcast keys in a wireless network of $|\mathcal{N}| = 500$ nodes.

we compare the five node capture strategies to compromise all target traffic in the network.

The simulation was performed for a network of $|\mathcal{N}| = 500$ randomly deployed nodes with an average of 25 neighbors. The subsets $\mathcal{S}, \mathcal{D} \subseteq \mathcal{N}$ were randomly selected such that $|\mathcal{S}| = 100$ and $|\mathcal{D}| = 10$. Each source node chose to route each message to the nearest three of the 10 destination nodes. Fig. 8 illustrates the performance of the node capture attack. As seen in Fig. 8, the use of the GNAVE algorithm with the route vulnerability metric $h_{\mathcal{C}}(s,d)$ requires the capture of significantly fewer nodes compared to the first four attacks. For example, to compromise 80% of network traffic, the five attack strategies require 36, 18, 16, 16, and 12 captured nodes. Hence, the total resource expenditure due to the capture of nodes in $\mathcal{C}$ using the route vulnerability metric $h_{\mathcal{C}}(s,d)$ is $35-75\%$ of that required by the first four simulated strategies.

VI. CONCLUSION

In this work, we analyzed the impact of node capture attacks on the confidentiality and integrity of network traffic. We mapped the compromise of network traffic to the flow of current through an electric circuit and proposed a new metric of route vulnerability that quantifies the resilience of traffic to the compromise of symmetric keys. We formulated the minimum cost node capture attack as a nonlinear integer programming minimization problem using the route vulnerability metric and provided a greedy heuristic solution called GNAVE to approximate the NP-hard minimization problem. We showed that an adversary can significantly decrease the resource expenditure by intelligently capturing nodes using the proposed route vulnerability metric. Our future work will include probabilistic estimation of route vulnerability when information about the key assignment and routing protocols is non-deterministic.

REFERENCES

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC, 1996.

[2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conference on Computer and Communications Security (CCS'02)*, Washington, DC, USA, Nov. 2002, pp. 41–47.

[3] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. 2003 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2003, pp. 197–213.

[4] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Information and System Security*, vol. 8, no. 2, pp. 228–258, May 2005.

[5] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Information and System Security*, vol. 8, no. 1, pp. 41–77, Feb. 2005.

[6] P. Tague and R. Poovendran, "A canonical seed assignment model for key predistribution in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 4, Oct. 2007.

[7] ——, "Modeling adaptive node capture attacks in multi-hop wireless networks," *Ad Hoc Networks*, vol. 5, no. 6, pp. 801–814, Aug. 2007.

[8] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. 2002 IEEE International Symposium on Information Theory (ISIT'02)*, Lausanne, Switzerland, Jun./Jul. 2002, p. 323.

[9] K. Jain, "Security based on network topology against the wiretapping attack," *IEEE Wireless Communication*, vol. 11, no. 1, pp. 68–71, Feb. 2004.

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press, McGraw-Hill, 2000.

[11] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, May 2005.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[13] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE International Symposium on Information Theory (ISIT'03)*, Yokohama, Japan, Jun./Jul. 2003, p. 441.

[14] R. Diestel, *Graph Theory*, 3rd ed. Springer, 2005.

[15] A. Bloch, "On methods for the construction of networks dual to non-planar networks," *Proceedings of the Physical Society*, vol. 58, no. 6, pp. 677–694, Nov. 1946.

[16] G. Dobson, "Worst-case analysis of greedy heuristics for integer programming with nonnegative data," *Mathematics of Operations Research*, vol. 7, no. 4, pp. 515–531, Nov. 1982.

[17] V. Chvatel, "Greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, Aug. 1979.